PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-183891
(43)Date of publication of application : 21.07.1995

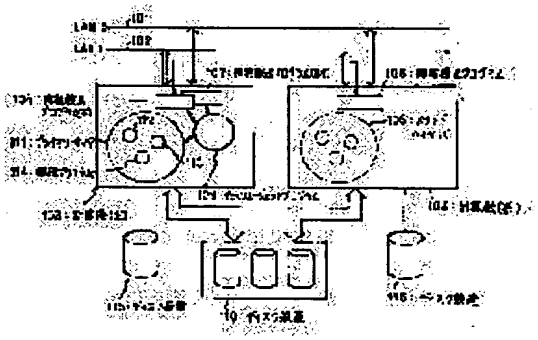
(51)Int.Cl. H04L 12/28
G06F 11/20
G06F 15/16

(21)Application number : 05-328162
(22)Date of filing : 24.12.1993

(71)Applicant : MITSUBISHI ELECTRIC CORP
(72)Inventor : UEMURA HOZE
SAKAKURA TAKASHI

(54) COMPUTER SYSTEM

(57)Abstract:
PURPOSE: To improve the working efficiency with less hardware development and to improve the reliability in a computer system.
CONSTITUTION: In a computer system composed of plural computers connected with one or plural networks, programs to be inspected 112 to 114 transmit detection signals to a fault detection program 108, 107 or 106, changing the transmission interval of the detection signals by the load status of a computer 103. A fault detection program also investigates the load status of the computer 103, checks whether the transmitted detection signals are time-out or not and judges the presence or absence of the generation of a fault.



LEGAL STATUS

[Date of request for examination]	07.08.1997
[Date of sending the examiner's decision of rejection]	17.12.2002
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	3447347
[Date of registration]	04.07.2003
[Number of appeal against examiner's decision of rejection]	2003-00947
[Date of requesting appeal against examiner's decision of rejection]	16.01.2003

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平7－183891

(43)公開日　平成７年(1995)７月21日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｈ０４Ｌ	12/28				
Ｇ０６Ｆ	11/20	３１０　Ｆ			
	15/16	４７０　Ｓ			
			7831－5Ｋ	Ｈ０４Ｌ　11/00　３１０　Ｄ	

審査請求　未請求　請求項の数13　ＯＬ　（全 14 頁）

(21)出願番号　特願平5－328162

(22)出願日　平成５年(1993)12月24日

(71)出願人　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目２番３号

(72)発明者　ウエムラ　ホゼ
鎌倉市大船五丁目１番１号　三菱電機株式会社情報システム研究所内

(72)発明者　坂倉　隆史
鎌倉市大船五丁目１番１号　三菱電機株式会社情報システム研究所内

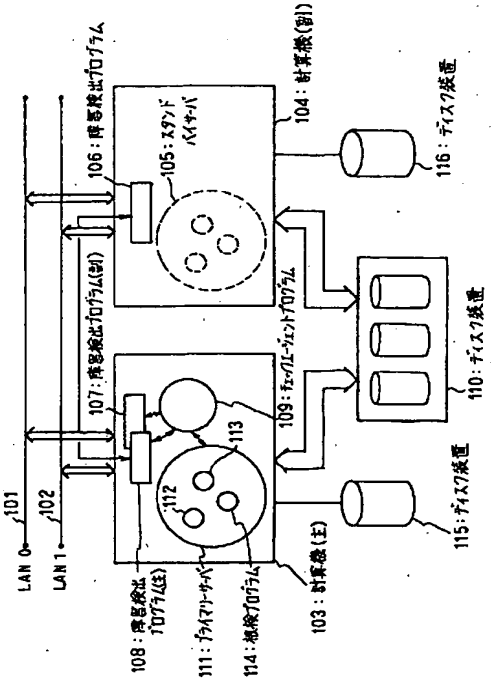
(74)代理人　弁理士　高田　守

(54)【発明の名称】　計算機システム

(57)【要約】

【目的】　計算機システムにおいて、より少ないハードウェア開発で稼働率の向上、ならびに、信頼性の向上を目的とする。

【構成】　１つ、あるいは、複数のネットワークに接続された複数の計算機から構成される計算機システムで、被検プログラム１１２，１１３，１１４は計算機１０３の負荷状況により検出信号の送出間隔を変えながら検出信号を障害検出プログラム１０８，１０７または１０６へ送出する。一方、障害検出プログラムも計算機１０３の負荷状況を調べ、送られてきた検出信号がタイムアウトか否かチェックし障害発生の有無を判断する。



1

【特許請求の範囲】

【請求項1】　（ａ）所定の処理を実行するとともに、その処理の実行中に障害の発生の有無を検出し検出信号として送信する被検ソフトウェア、（ｂ）上記被検ソフトウェアからの検出信号を受信して、被検ソフトウェアの障害を検出する障害検出ソフトウェアを有する計算機システムにおいて、

上記被検ソフトウェア及び障害検出ソフトウェアは、それぞれシステムの負荷を検出する負荷検出部を備え、負荷検出部により検出したシステムの負荷に応じて検出信号の送受信間隔を調整することを特徴とする計算機システム。

【請求項2】　（ａ）所定の処理を実行するとともに、その処理の実行中に障害の発生の有無を検出し検出信号として送信する被検ソフトウェア、（ｂ）上記被検ソフトウェアからの検出信号を受信して、被検ソフトウェアの障害を検出する障害検出ソフトウェアを有する計算機システムにおいて、

上記被検ソフトウェアは、検出信号の送出間隔を上記障害検出ソフトウェアに通知し、障害検出ソフトウェアは通知された検出信号の送出間隔に基づいて被検ソフトウェアの障害を検出することを特徴とする計算機システム。

【請求項3】　上記被検ソフトウェアは、システムの負荷を検出する負荷検出部を備え、負荷検出部により検出したシステムの負荷に応じて検出信号の送出間隔を調整することを特徴とする請求項2 記載の計算機システム。

【請求項4】　上記被検ソフトウェアは、複数のプログラムを備え、各プログラムが検出信号を送信するとともに、

上記障害検出ソフトウェアは、上記複数のプログラムの中の関連するプログラムを示す管理情報を記憶するとともに、管理情報に基づき関連する複数のプログラムからの検出信号を受信して関連する複数のプログラムの障害を検出することを特徴とする請求項1，２または3 記載の計算機システム。

【請求項5】　上記障害検出ソフトウェアは、第1 の障害検出ソフトウェアと第2 の障害検出ソフトウェアを備えており、一方の障害検出ソフトウェアが他方の障害検出ソフトウェアの障害を検出することを特徴とする請求項1，２または3 記載の計算機システム。

【請求項6】　上記被検ソフトウェアは、障害検出ソフトウェアに対する動作手順を指示するメッセージを検出信号中に含ませるとともに、上記障害検出ソフトウェアは、手順情報を持ち、メッセージと手順情報により指示された手順を実行することを特徴とする請求項1，２または3 記載の計算機システム。

【請求項7】　上記計算機システムは、複数の計算機を備えており、各計算機は各計算機の状態を調査して上記障害検出ソフトウェアに出力する調査プログラムを備え、上記障害検出ソフトウェアは調査プログラムからの調査結果に基づいて、障害が検出された被検ソフトウェアを他の計算機により実行させることを特徴とする請求項1，２または3 記載の計算機システム。

2

【請求項8】　主記憶装置および2 次記憶装置を持つ複数の計算機が接続された計算機システムにおいて、主記憶装置の内容を分割して複数の計算機の2 次記憶装置に退避する分割退避手段を備えたことを特徴とする計算機システム。

【請求項9】　上記計算機システムは、あらかじめ、分割退避先を設定した管理表を備えたことを特徴とする請求項8 記載の計算機システム。

【請求項１０】　上記計算機システムは、さらに、ネットワークに2 次記憶装置を持つ複数の計算機が接続され、上記分割退避手段は、ネットワークを用いて複数の２次記憶装置にデータを退避することを特徴とする請求項8 記載の計算機システム。

【請求項１１】　以下の要素を有する計算機システム（ａ）データを記憶する２次記憶装置、（ｂ）上記データを一時的に記憶する主記憶装置、（ｃ）上記主記憶装置に記憶されたデータの2 次記憶装置への書き戻しの有無と、そのデータの正当性を判断するデータを管理情報として記憶する記憶手段、（ｄ）障害発生時に、上記記憶手段により記憶された管理情報により上記主記憶装置に記憶されたデータのうち、２次記憶装置に書き戻されていないデータを判定し、その正当性をチェックしてデータを書き戻す書き戻し手段。

【請求項１２】　上記計算機システムは、複数の2 次記憶装置を備えた多重系システムであり、上記書き戻し手段は、複数の2 次記憶装置に対してデータを書き戻すことを特徴とする請求項1 1 記載の計算機システム。

【請求項１３】　以下の要素を有する計算機システム（ａ）データを記憶する２次記憶装置、（ｂ）上記2 次記憶装置に対してデータを書込む書込み手段、（ｃ）上記書込み手段により更新しようとする時、元のデータを不揮発性記憶媒体に退避し、更新開始を示すマークと、更新完了後に更新完了のマークを退避データにマークする退避手段、（ｄ）障害発生時に、上記退避手段により退避したデータを調べ、更新開始を示すマークがあり、更新完了のマークがないデータをデータの更新中にエラーが発生したとみなし、退避データを用いて上記2 次記憶装置のデータを復旧する復旧手段、（ｅ）更新成功時には退避データを開放する開放手段。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、耐故障性を備えた計算機システムに関し、特に、ネットワークに接続された、複数台の計算機から構成される計算機システムに関する。

【０００２】

3

【従来の技術】近年、計算機システムの信頼性に対する関心が高まってきている。その背景として、従来の計算機システムにおいて、ダウンサイジングとよばれるように汎用機上にあった機能を端末側に移行する、また、センタ機能自体をマイクロプロセッサベースの計算機上で実現することによる低コスト化を図る試みが盛んになされている。ところが、汎用機システムがハードウェアの故障、誤動作、また、これらを要因とするソフトウェア障害、ソフトウェア自体の障害に対してその対処手段を持ち、より高い稼働率や信頼性を実現するのに対し、マイクロプロセッサベースの計算機システムでは、障害対処手段が貧弱であり、高い稼働率や、信頼性を要求されるアプリケーションを稼働させるのは難しかった。

【０００３】ところで、ダウンサイジングにあたって、特徴的なこととして、計算機の接続形態の変化がある。従来の汎用機を中心としたシステムが、その端末群と、電話回線や、シリアルラインによってクラスタ接続されていたのに対し、主にほぼ等価なマイクロプロセッサベースの計算機で構成されるシステムでは個々の計算機は、高速なローカルエリアネットワークに接続されることが多くなった。

【０００４】本発明の課題である計算機システムの高い稼働率、高信頼性を実現するために、過去に商用化されたものとして、タンデム社、ストラタス社、セコイアシステム社などによるものがある。これらのシステムは、ハードウェアのコンポーネントを冗長化し、シングルポイントフェイル、つまり、ある一つのコンポーネントが故障したためにシステム全体が停止しないようにできている。しかし、ハードウェアの冗長化のため、これらシステムは高価なものとなっている。

【０００５】近年のハードウェアの信頼性の向上、また、要求される稼働率や信頼性のレベルにより、必ずしも全てのコンポーネントの冗長化は、コスト面も考えるとそれを要求できない場合も多い。本発明は、かかるシステムに適用され、稼働率、信頼性の向上を図るもので、ハードウェアの機構を最小限にとどめ、主にソフトウェアによる実現を図る。ハードウェアの冗長度が低い計算機群で、これらの課題を達成するにあたって、高速なローカルエリアネットワークが鍵となる。つまり、ハードウェアの冗長度に欠けるところを、ネットワークを通して、複数台の計算機で補い合うことでカバーする。このようなアプローチによる製品も幾つか市場に既に出ていて、ＤＥＣ社のＶＡＸクラスタ、ＩＢＭ社のＨＡＮＦＳなどがある。ＤＥＣ社のＶＡＸクラスタと呼ばれる製品は、デュアルポートディスクを共有する、ネットワークで接続された複数台の計算機で構成され、主系の計算機に障害が発生した時は、従系の計算機が業務を代行する。ＩＢＭ社のＨＡＮＦＳは、同様にディスク装置を共有するネットワークに接続された複数の計算機から構成される計算機上で実行されており、複数の計算機間で

4

ファイル共有サービスを行うプログラムに障害が発生した時は、該サービスプログラムは、ディスク装置を共有している他の計算機上で起動される。

【０００６】特開平4 －２３０５３８には、一定時間間隔内に応答が受信されるか否かを障害発生の判断の1つとした障害ソフトウェアコンポーネントの検出方法、特開平4 －３４０６４９には、検知信号の発信によらないソフトウェア障害の検出方法が述べられている。特開平１－６１８５５には、マルチプロセッサシステムにおけるバックアッププロセッサの起動方法が述べられている。

【０００７】また、障害発生時の主記憶上のデータのダンプの高速化を図るものの先行事例としては、特開平3 －２１１６３８に、コアデータを圧縮した上で2次記憶上に退避する方法が述べられている。また、ディスクデータの一貫性を図るために考えられた先行特許として、特開平1 －２７７３７２では、エラー発生すると書き込んだデータ内容を、ホストシステムに返送する方式が述べられている。

【０００８】

【発明が解決しようとする課題】本発明の課題は、より少ないハードウェアの投資で、高い稼働率や、信頼性を提供する計算機システムを構築するための、基本的要素を提供するものである。

【０００９】従来のシステムにおいて、ソフトウェア障害検知機構がタイムアウトを、その障害発生状態か否かを判断する基準にしている場合、計算機の負荷状態によっては送信のために決められた時間よりも長い時間がかかってしまい、タイムアウトとなり、障害発生と判断され、正確な判断が下せないという問題点があった。

【００１０】また、障害検知しようとするプログラムが幾つかのプログラムのサービスを利用して成り立っているとき、あるいは、相互にサービスを利用しあって成り立っているとき、目的とするプログラムの障害検知を行うだけでは不十分で、正確な判断ができないという問題点があった。

【００１１】そして、従来の障害検知方式では、検知機構自体のシングルポイントフェイルに対応できないという問題点があった。

【００１２】更に、従来の障害検知、復旧方式では、復旧手段が一律的であり、障害に対して、必要以上の処置をとらざるを得ない場合が多いという問題点があった。

【００１３】また、ソフトウェア障害が発生し、その復旧にあたって、ネットワーク内の他の計算機上に移行する必要が生じたときに、固定的に移行先の計算機を決めたのでは、移行先の負荷状況、資源状況によって、必ずしも移行先として望ましいものとはならない場合があるという問題点があった。

【００１４】より高稼働率、高信頼性システムを構築する上で、障害発生時になくてはならない、主記憶データ

5

の採取を高速に行うことも課題である。稼働率は、故障修理期間を短くすることによって向上できる。従って、システム再立ち上げを行う場合に、主記憶データのダンプ時間を短くすることは、稼働率向上に寄与する。

【００１５】また、主記憶上にキャッシュされたディスクデータの一貫性を維持することを課題としており、これによりシステム再立ち上げの場合、ディスク上に構築されたファイルシステムの一貫性回復のために要する時間を最小に押さえることができる。

【００１６】また、データベースプログラムなどの保証すべき信頼性が特に高いアプリケーションに利用されるべき機能で、全てのディスクに対する書き込みオペレーションに対し、成功か不成功かの場合に、更新後、更新前の状態を必ず保証することにより、従来アプリケーションプログラムの中で行っていたデータ一貫性保持操作を簡略化し、かつ、ハードウェアレベルでデータの一貫性を保証し、システムの高速化を図ることも課題である。

【００１７】本発明は、上記のような問題点を解決し、課題を達成するためになされたもので、より少ないハードウェア開発で、より高い計算機システムの稼働率、信頼性を実現することを目的とする。また、障害復旧時の計算機立ち上げ時間を短縮することにより、稼働率の向上を図ることを目的とする。また、ディスクデータの一貫性を保証して、アプリケーションプログラムの中で行っていた、データ一貫性保持操作を簡略化することにより、システムの高速化を図ることが目的である。

【００１８】

【課題を解決するための手段】第1の発明に係る計算機システムは、（ａ）所定の処理を実行するとともに、その処理の実行中に障害の発生の有無を検出し検出信号として送信する被検ソフトウェア、（ｂ）上記被検ソフトウェアからの検出信号を受信して、被検ソフトウェアの障害を検出する障害検出ソフトウェアを有する計算機システムにおいて、上記被検ソフトウェア及び障害検出ソフトウェアは、それぞれシステムの負荷を検出する負荷検出部と、システムの負荷に応じて検出信号の送受信間隔を調整するための頻度表を持つことを特徴とする。

【００１９】第2の発明に係る計算機システムは、（ａ）所定の処理を実行するとともに、その処理の実行中に障害の発生の有無を検出し検出信号として送信する被検ソフトウェア、（ｂ）上記被検ソフトウェアからの検出信号を受信して、被検ソフトウェアの障害を検出する障害検出ソフトウェアを有する計算機システムにおいて、上記被検ソフトウェアは、検出信号の送出間隔の情報を検出信号にふくませ上記障害検出ソフトウェアに通知し、障害検出ソフトウェアは通知された検出信号の送出間隔に基づいて被検ソフトウェアの障害を検出することを特徴とする。

【００２０】第3の発明に係る計算機システムは、上記

6

第2の発明に係る計算機システムにおいて、上記被検ソフトウェアにシステムの負荷を検出する負荷検出部を備え、負荷検出部により検出したシステムの負荷に応じて検出信号の送出間隔を調整するための頻度表を持つことを特徴とする。

【００２１】第4の発明に係る計算機システムにおいて、上記被検ソフトウェアは、複数のプログラムを備え、各プログラムが検出信号を送信するとともに、上記障害検出ソフトウェアは、上記複数のプログラムの中の関連するプログラムを示す管理情報を記憶するとともに、管理情報に基づき関連する複数のプログラムからの検出信号を受信して関連する複数のプログラムの障害を検出することを特徴とする。

【００２２】第5の発明に係る計算機システムにおいて、上記障害検出ソフトウェアは、第1の障害検出ソフトウェアと第2の障害検出ソフトウェアを備えており、一方の障害検出ソフトウェアが他方の障害検出ソフトウェアの障害を検出することを特徴とする。

【００２３】第6の発明に係る計算機システムにおいて、上記被検ソフトウェアは、障害検出ソフトウェアに対する動作手順を指示するメッセージを検出信号中に含ませるとともに、上記障害検出ソフトウェアは、メッセージに対応する手順を記載した手順情報を持ち、メッセージと手順情報により指示された手順を実行することを特徴とする。

【００２４】第7の発明に係る計算機システムは、複数の計算機を備えており、各計算機は各計算機の状態を調査して上記障害検出ソフトウェアに出力する調査プログラムを備え、上記障害検出ソフトウェアは調査プログラムからの調査結果と被検ソフトウェアの実行条件を考慮して、障害が検出された被検ソフトウェアを他の計算機により実行させることを特徴とする。

【００２５】第8の発明に係る計算機システムは、主記憶装置および2次記憶装置を持つ複数の計算機が接続された計算機システムにおいて、主記憶装置の内容を分割して複数の計算機の2次記憶装置に退避する分割退避手段を備えたことを特徴とする。

【００２６】第9の発明に係る計算機システムは、上記第8の発明に係る計算機システムにおいて、あらかじめ、分割退避先を設定した管理表を備えたことを特徴とする。

【００２７】第１０の発明に係る計算機システムは、上記第8の発明に係る計算機システムにおいて、さらに、ネットワークに2次記憶装置を持つ複数の計算機が接続され、上記分割退避手段は、ネットワークを用いて複数の2次記憶装置にデータを退避することを特徴とする。

【００２８】第１１の発明に係る計算機システムは、以下の要素を有することを特徴とする。（ａ）データを記憶する2次記憶装置、（ｂ）上記データを一時的に記憶する主記憶装置、（ｃ）上記主記憶装置に記憶されたデ

7

ータの2 次記憶装置への書き戻しの有無と、そのデータの正当性を判断するデータを管理情報として記憶する記憶手段、（ｄ）障害発生時に、上記記憶手段により記憶された管理情報により上記主記憶装置に記憶されたデータのうち、2 次記憶装置に書き戻されていないデータを判定し、その正当性をチェックしてデータを書き戻す書き戻し手段。

【００２９】第１２の発明に係る計算機システムは、上記第１１の発明に係る計算機システムにおいて、複数の2 次記憶装置を備えた多重系システムであり、上記書き戻し手段は、複数の2 次記憶装置に対してデータを書き戻すことを特徴とする。

【００３０】第１３の発明に係る計算機システムは、以下の要素を有することを特徴とする。（ａ）データを記憶する2 次記憶装置、（ｂ）上記2 次記憶装置に対してデータを書込む書込み手段、（ｃ）上記書込み手段により更新しようとする時、元のデータを不揮発性記憶媒体に退避し、更新開始を示すマークと、更新完了後に更新完了のマークを退避データにマークする退避手段、（ｄ）障害発生時に、上記退避手段により退避したデータを調べ、更新開始を示すマークがあり、更新完了のマークがないデータをデータの更新中にエラーが発生したとみなし、退避データを用いて上記2 次記憶装置のデータを復旧する復旧手段、（ｅ）更新成功時には退避データを開放する開放手段。

【００３１】

【作用】第1 の発明における計算機システムは、被検ソフトウェアが所定の処理実行中に障害の発生を検出し、検出信号として送信し、障害検出ソフトウェアが検出信号を受信して被検ソフトウェアの障害を検出する。このように2 つの独立したプログラムとすることにより、検出信号が所定の間隔で到着しない場合、障害検出ソフトウェア側は被検ソフトウェア側で検出信号を送信できない障害が発生したとみなして、タイムアウトを障害発生か否かの判断の1 つにすることができる。このとき、検出信号の到着に要する時刻は、システムの負荷によって遅れることがあるため、双方のソフトウェアにシステムの負荷を検出する負荷検出部を持たせ、負荷に応じて検出信号の送受信間隔を調整する。

【００３２】第2 の発明における計算機システムは、被検ソフトウェアが検出信号の送出間隔を障害検出ソフトウェアに通知するため、障害検出ソフトウェア側で送出間隔を調べる手順を省くことができる。

【００３３】第3 の発明における計算機システムは、被検ソフトウェア側に負荷検出部と頻度表を持たせ、負荷により検出信号の送出間隔の調整を行う。また、その送出間隔の情報も検出信号中に含ませ障害検出ソフトウェア側に通知するので、障害検出ソフトウェア側では、負荷検出部や頻度表による処理を行う必要がない。

【００３４】第4 の発明における計算機システムでは、

8

上記被検ソフトウェアは、複数のプログラムを備え、各プログラムが検出信号を送信する。上記障害検出ソフトウェアは、上記複数のプログラムの中の関連するプログラムを示す管理情報を記憶し、管理情報に基づき関連する複数のプログラムからの検出信号を受信して、関連する複数のプログラムの障害を検出することができる。

【００３５】第5 の発明における計算機システムは、障害検出ソフトウェアの障害を検出するソフトウェアを備えることにより、障害検知機構自体のシングルポイントフェイルを回避できる。

【００３６】第6 の発明における計算機システムは、上記被検ソフトウェアが、障害検出ソフトウェアに対する動作手順を指示するメッセージを検出信号中に含ませる。上記障害検出ソフトウェアは、メッセージに対応する手順を記した手順情報を持ち、手順情報によりメッセージに対応する手順を実行するため、無駄な処理を省くことができる。

【００３７】第7 の発明における計算機システムでは、上記計算機システムは、複数の計算機を備えており、各計算機は各計算機の状態を調査して上記障害検出ソフトウェアに出力する調査プログラムを備える。上記障害検出プログラムは被検ソフトウェアの実行条件を考慮して調査プログラムから得られた負荷状況、資源状況に基づき、障害が検出され、かつ他の計算機への移行が必要となった被検ソフトウェアの移行に最適な計算機を探し、移行して実行する。

【００３８】第8 の発明における計算機システムは、障害発生時に主記憶装置の内容を分割して、接続されている複数の他の計算機の2 次記憶装置に退避することができる。

【００３９】第9 の発明における計算機システムは、分割退避先を設定した管理表を備えているので、主記憶装置のデータの採取先をすみやかに決めることができる。

【００４０】第１０の発明における計算機システムは、ネットワークを介して分割されたデータを退避する作業を他の計算機に分割することができる。

【００４１】第１１の発明における計算機システムは、主記憶装置上にキャッシュされたデータの2 次記憶装置への書き戻しの有無と、そのデータの正当性を判断するデータを管理情報として持つ。これにより、障害発生時にこの管理情報に基づいて主記憶装置に記憶されたデータのうち、2 次記憶装置に書き戻されていないデータを判断し、またそのデータに障害の影響があるかどうか正当性を再度チェックし、データを書き戻す。

【００４２】第１２の発明における計算機システムは、多重系システムにおいても上記第１１の発明と同様に、障害発生時の主記憶装置上に読み出され、変更されたデータとディスクデータの一貫性を保つことができる。

【００４３】第１３の発明における計算機システムは、ディスク装置への書き込みの際に、更新される前のデー

9

タに更新開始のマークをしてそのデータを退避した後、ディスク装置への書き込みを開始し、更新終了後、退避したデータに更新完了のマークをする。ディスクへの書き込み命令が異常終了した場合に、この退避したデータの中で更新完了のマークがない退避データを書き戻すことにより、書き込み命令発行以前のデータ状態を保証することができる。また、更新成功時には退避データを開放する。

【００４４】

【実施例】

実施例1．本実施例における計算機システムの例を図1を用いて説明する。１０３および１０４は独立した計算機で、ネットワーク１０１，１０２にそれぞれ接続されている。図中には計算機は2台しか描かれていないが、台数に制限はない。主／副形態で、機能の冗長性を実現する場合は、主／副それぞれの計算機からアクセスできるディスク装置１１０により、二つの計算機間でのコンシステントなデータの引渡しを可能にする。プライマリーサーバ１１１にあるプログラム群１１２，１１３，１１４はそれぞれ依存関係を持つアプリケーションプログラム、つまり、被検プログラムに相当する。１０６，１０７，１０８は障害検出プログラムで、被検プログラムが実行されている同一計算機上で、また、ネットワーク内の違う計算機上で実行される。１０９は、これら障害検知機構のセクションの開始／終了などのサービスを行うチェックエージェントプログラムである。

【００４５】障害検知機構は、アプリケーションプログラム（被検プログラム）の中で処理される部分と障害検出プログラムの中で処理される部分に分かれて存在している。図2は本実施例による障害検知機構のソフトウェア構成を示している。被検プログラム２０１は、計算機の負荷状況を調査する負荷検出部２０２と、図3にその内容を示す負荷に対しての障害検知頻度を示す頻度表２０３と、障害検出信号を障害検出プログラムに送出する送信部２０４を含んでいる。一方、障害検出プログラムは、２０２に等価な負荷検出部２０６，２０３に等しい頻度表２０７、被検プログラムからの障害検出信号を受信する受信部２０８を含んでいる。被検プログラム２０１は、一定期間中に何回かの割合で送信部２０４から”私は正常である”旨のメッセージを障害検出プログラム２０５に送信している。障害検出プログラム２０５の受信部２０８はそのメッセージを受け取り、その内容またはメッセージが到着するか否かで被検プログラムが正常であるか否かを判断している。

【００４６】そのため、ソフトウェア障害検知機構がタイムアウト（次のメッセージが到着すべき時刻に到着しない場合に異常であると判断するまでの時間）を障害発生状態か否かを判断する基準にしているので、計算機の負荷状態によっては、一律な判断基準では正確な判断が下せないという問題点があった。そこで、双方のプログラムは、計算機の負荷状態を定期的に採取し、その値から、障害検知頻度を頻度表に従って設定する。例えば、頻度表の内容が図3のよう場合、負荷が0のときは頻度は１０であるから、双方のプログラムは一定期間中に１０回障害検知のためのメッセージのやり取りを行うことになる。なお、負荷は計算機の稼動率とジョブキューの長さで決まる。

10

【００４７】また、なぜ双方のプログラムで、負荷検出部と頻度表を持つかを説明する。被検プログラムで、計算機の負荷状態により送信頻度を変えているため、障害検出プログラム側のタイムアウト値も変える必要がある。そのため、障害検出プログラム側でも被検プログラムと同様の処理を行い、新しいタイムアウト値を設定する。

【００４８】被検プログラムと障害検出プログラムの中の負荷検出部は、オペレーティングシステムの問い合わせ手段を用い、ジョブキューの長さと計算機の稼動率により得る。ただし計算機の負荷状況は急峻に変化することがあるので、双方の負荷検出値に差異が生じることを防ぐために、負荷検出部には十分長いサンプリング期間を持たせる。負荷状況はたえず変化するものなので、負荷は一瞬一瞬の細かい値の検出ではなく、その時間帯の大まかな傾向値とした方がより実際的である。

【００４９】また、被検プログラムと障害検出プログラムに分けているのはプログラムの作り勝手によるためと、独立したプログラムにすることにより、被検プログラム側で検出信号を送信できない状態に陥った場合、障害検出プログラム側ではタイムアウトを障害発生か否かの判断の1つにするためである。

【００５０】以上のようにこの実施例では、該システム上で走行するソフトウェアに発生する障害を検知し、障害状態から回復せしめることを特徴とする計算機システムであって、該障害検知に用いる、被検ソフトウェア自身の送出する、検出信号の送出頻度を、該システムの負荷状況により調整する。

【００５１】そのために計算機システムの負荷状況検出手段、および、計算機システムの負荷状況と、障害検知頻度の頻度表を、検知信号送信側、つまり、被検プログラムと検知機構にそれぞれ持つことによって、計算機システムの負荷状況によって、障害検知頻度の調整を行うようにしている。

【００５２】実施例2．この実施例では、被検プログラム側の障害検知信号中に、障害検知信号の送信間隔情報（インターバル情報ともいう）を送り、それをもとに、障害検出プログラムがタイムアウト時間を設定する例について述べる。

【００５３】図4はこの実施例の障害検知機構のソフトウェア構成を示している。被検プログラム４０１は、計算機の負荷状況を調査する負荷検出部４０２と、図5にその内容を示す負荷に対しての障害検知の頻度と、障害

検出プログラムへ送信する障害検知のインターバル情報を示す頻度表４０３と、障害検出信号を障害検出プログラムに送出する送信部４０５を含んでいる。一方、障害検出プログラム４０６は、被検プログラムからの障害検出信号を受信する受信部４０７を含んでいる。被検プログラムは、計算機の負荷状態を定期的に採取し、その値から頻度表に従って障害検知頻度を設定し、また、頻度表から障害検知信号の一部として送るべき送出情報を設定する。例えば、図5 の５０１ならば、負荷が0 の時は一定期間に１０回の頻度で、被検プログラムは送出情報１を含む障害検知信号を、障害検出プログラムに送信する。障害検出プログラムは、タイムアウト値を送出情報に合わせて設定して、障害か否かの判断を行う。

【００５４】これは被検プログラムが正常か否かを障害検出プログラムにおいて、ある一定期間内に次のメッセージが到着するか否かでも判断しているためである。また、被検プログラム側で負荷の状況により、頻度表を参照して検出信号の送信間隔を変えているので、その値を送出情報として障害検出プログラムに知らせる。これにより障害検出プログラム側では送出情報により被検プログラム側での変化に合わせてタイムアウト値の変更を行うことができる。

【００５５】以上のようにこの実施例では、障害検出信号に検知頻度調整のための情報を付加することを特徴としている。

【００５６】実施例3．障害検知しようとするプログラムが幾つかのプログラムのサービスを利用して成り立っているとき、あるいは、相互にサービスを利用しあって成り立っているとき、目的とするプログラムの障害検知を行うだけでは不十分で正確な判断ができない。そのためこの実施例では、監視すべきプログラム、および、監視すべきプログラムがそのサービスを利用しているプログラムとのプログラム間の依存関係を示す表を、障害検出機構に持つことにより、上記依存関係を持つプログラムの監視を可能にする例について述べる。

【００５７】被検プログラム、障害検出プログラムは、実施例1 または、実施例2 の機能を持つ。図6 は計算機システム上で、ある瞬間のプログラムの実行状況を示した図である。アプリケーションプログラムＡ（６０２）、アプリケーションプログラムＢ（６０３）、アプリケーションプログラムＣ（６０１）、障害検出プログラム６０４が実行されている。障害検出プログラム６０４は、図7 に詳細を示すアプリケーションプログラム間の依存関係表６０５、各アプリケーションからの障害検出信号を受信する受信部６０６を含んでいる。

【００５８】この依存関係表６０５は、つぎのようにして設定する。例えば、プログラムＡを作る時、依存するプログラムはプログラムＢとＣであると判る。プログラムＢは、プログラムＡに依存しており、またプログラムＡをとおしてプログラムＣに依存している。プログラムＣは、どのプログラムにも依存していない。このように、各プログラム間の依存関係がわかるので、障害検出プログラムを作成するときにこれを依存関係表６０５として持たせる。

【００５９】図7 は、アプリケーションプログラム間の依存関係を表す依存関係表で、例えば、ＡはＣとＢに依存しており（７０１）、ＢはＡ、Ａはさらにに依存しており（７０２）、Ｃはサービスは提供するがいずれのプログラムにも依存していない（７０３）ことを示すものである。障害検出プログラム６０４は、この依存関係表を参照し、例えばＡのプログラムをモニタする場合には、ＣおよびＢのプログラムの障害検知も行う。

【００６０】このようにアプリケーションプログラム間の依存関係表を持つことにより監視すべきプログラムおよびこのプログラムが利用するプログラムを総合的に監視することが可能になる。

【００６１】以上のように、この実施例では、該システム上で互助動作する複数のソフトウェアに発生する障害を検知し、障害状態から回復することを目的とし、該複数ソフトウェアの管理情報を持ち、該管理情報中に記述される全てのソフトウェアについて、障害検出、および、障害回復を行うことを特徴とする計算機システムについて述べた。

【００６２】実施例4．今までの障害検知方式では、検知機構自体のシングルポイントフェイルに対応できなかった。この実施例では障害検知機構を2 重化することにより、障害検知機構自体の障害による、システム障害を回避する例を説明する。

【００６３】図8 は被検プログラムと障害検出プログラムのソフトウェア構成を示した図である。被検プログラム８０１は、障害検知信号を主障害検出プログラム８０２、および、副障害検出プログラム８０３に送信する。図9 に副障害検出プログラムの動作を示す。もし障害が検出されたならば（９０１）、副障害検出プログラムは主障害検出プログラムの状態をチェックし（９０２）、健全ならば何もしない。もし健全でなければ、障害検出プログラム復旧（９０３）を行う。障害検出プログラム復旧とは、副障害検出プログラムが主障害検出プログラムを停止させ、副障害検出プログラムが主障害検出プログラムの代わりに被検プログラムの障害に対処する。また、この時副障害検出プログラムは、自分の複製を作り、以後これに自分を監視させる。なお、当実施例において主障害検出プログラムは、副障害検出プログラムの存在を意識しない。

【００６４】この実施例では、該システム上で走行するソフトウェアに発生する障害を検知し、障害状態から回復せしめることを特徴とする計算機システムであって、障害検知機構を2 重化することにより、障害検知機構自体の障害による、システム障害を回避することを特徴とする計算機システムについて述べた。

13

【００６５】実施例5．上記実施例は1 例に過ぎず、副障害検出プログラムが、主障害検出プログラムのみを監視する方式もある。図１０はこの実施例の被検プログラムと障害検出プログラムの関係を示した図である。

【００６６】実施例6．従来の障害検知、復旧方式では、復旧手段が一律的であり、障害に対して、必要以上の処置をとらざるを得ない場合が多かった。この実施例では、被検ソフトウェアの送出する検出信号によって障害種類を類別する障害検出機構を有し、障害種類によって障害回復手順を記述した手順情報を持つこと、また、手順情報の設定手段を持つことによって障害種類に応じた障害復旧手段を提供する例について述べる。

【００６７】この実施例は実施例1 から5 にある障害検知機構に適用されるもので、障害検知信号に応じて、障害復旧、もしくは、サービスを行う。図１１に、障害検知信号と、障害検出プログラムが起動するサービスの手順の対応を示す対応表を示す。障害検出プログラムは、そのプログラム内にこの対応表を含み、障害検知信号を受けとったならば、対応する手順を実行する。図１１について説明する。正常信号を受けとっている限り、障害検出プログラムは何もしない（１００１）。停止信号を受けとった時は、障害検出プログラムはタイムアウトを延期する（１００２）。開始信号を受けとった時は、障害検出プログラムは、該被検プログラムの監視を開始する（１００３）。終了信号を受けとった時は、被検プログラムの監視を終了または、終了処理を行う（１００４）。障害1 信号を受けとった時は、同じ処理を3 回リトライする（１００５）。障害2 信号を受けとった時は、ディスクデータを修復する（１００６）。障害3 信号を受けとった時は、他計算機で再実行する（１００７）。

【００６８】以上のようにこの実施例では、該システム上で走行するソフトウェアに発生する障害を検知し、障害状態から回復せしめることを特徴とする計算機システムであって、被検ソフトウェアの送出する検出信号によって障害種類を類別する障害検出機構を有し、障害種類によって障害回復手順を記述した手順情報を持つことを特徴とする計算機システムについて説明した。

【００６９】実施例7．ソフトウェア障害が発生し、その復旧にあたって、ネットワーク内の他の計算機上に移行する必要が生じた時に、固定的に移行先を決めたのでは移行先の負荷状況、資源状況によって、必ずしも移行先として望ましいものとはならない場合があった。そこで、この実施例ではあるサービスが実行されていた計算機に障害が起きた時に、ネットワーク内のどの計算機でサービスを継続するかを決定するシステムについて述べる。すなわち、ネットワーク内の各計算機の負荷状況、資源状況を表す表と、その更新手段と、起動すべきプログラムと、負荷状況、および、資源状況との対応を示す表と、負荷、資源状況の比較結果により、指定されたプログラムの起動を行うことによって達成される。

14

【００７０】この実施例が適用される計算機システムの例を図1 を用いて説明する。１０３および１０４は独立した計算機で、ネットワーク１０１，１０２にそれぞれ接続されている。図中には計算機は2 台しか描かれていないが、台数に制限はない。主／副形態で、機能の冗長性を実現する場合は、主／副それぞれの計算機からアクセスできるディスク装置１１０により、二つの計算機間でのコンシステントなデータの引渡しを可能にする。プライマリーサーバ１１１にあるプログラム群１１２，１１３，１１４はそれぞれ依存関係を持つアプリケーションプログラム、つまり、前述した実施例で述べてきた被検プログラムに相当する。１０６，１０７，１０８は障害検出プログラムで、被検プログラムが実行されている同一計算機上で、また、ネットワーク内の違う計算機上で実行される。１０９は、これら障害検知機構のセッションの開始／終了などのサービスを行うチェックエージェントプログラムである。

【００７１】もし、被検プログラムに障害が発生したとき、さらに、計算機１０３自体が稼働不能に陥ったときは、スタンドバイサーバ１０５のアプリケーションプログラム群は、ディスク装置１１０、または、ネットワークを通してディスク装置１１５のデータが複写されているディスク装置１１６からデータを引き継ぎ、起動される。この時、これらの引き継ぎ処理を行うのは、計算機１０４上の障害検出プログラム１０６である。

【００７２】この実施例は、あるアプリケーションプログラムが実行されている計算機が稼働不能に陥ったときに、いずれかの計算機で再実行される時に適用されるもので、図１２にそのソフトウェア構成を示す。障害検出プログラム１２０３は、ある計算機上で実行されている被検プログラム１２０１から障害検知信号を受けとり、また、ネットワークに接続された、各計算機の負荷、資源状況を調査するプログラム１２０２からの状況報告を定期的に受ける。障害検出プログラム１２０３は、被検プログラム１２０１の実行条件データをそのプログラムに含む。

【００７３】各計算機の負荷、資源状況を調査するプログラム１２０２は、オペレーティングシステムの問い合わせ手段を用い、各計算機の負荷、資源状況を調査する。計算機の負荷や資源状況は、報告を受けた時点では変化していることもあるので、大まかな傾向がわかればよいと考え、十分長いサンプリング期間を持たせる。

【００７４】実行条件データの例を図１３に示す。プログラムＡは、計算機の負荷が1 以下で、Ｉ／Ｏ頻度が１００以下、主記憶残が2 以上というのがその実行の条件である（１１０１）。プログラムＢは、計算機の負荷が４以下で、Ｉ／Ｏ頻度が１０００以下、主記憶残が0．１以上というのが、その実行の条件である（１１０２）。一方、障害検出プログラム１２０３は、負荷、資

15

源状況を調査するプログラム１２０２から、図１４に示すような情報を定期的に受ける。計算機Ａは負荷が０．１でＩ／Ｏ頻度が１０、主記憶残が１００である（１３０１）。計算機Ｂは負荷が2で、Ｉ／Ｏ頻度が５０、主記憶残が１０である（１３０２）。計算機Ｃは負荷が1で、Ｉ／Ｏ頻度が１０００、主記憶残が５０である（１３０３）。これらの情報を照らし合わせた上で、障害検出プログラムは、障害が発生した被検プログラムをどの計算機上で再起動するかを決定する。例えば、図１３、および、図１４のデータで、プログラムＡに障害が発生したとすると、プログラムＡは計算機Ａ上で再起動される。

【００７５】以上のように、この実施例では、１つ、あるいは、複数のネットワークに、複数台接続された計算機によって構成され、互助動作する計算機システムにおいて、該システム上で走行するソフトウェアは、障害検知手段により監視され、該ソフトウェアが障害状態であり、かつ、走行中の計算機自体に障害があった時に、ネットワークリンク内の健全な他の計算機上で、該ソフトウェアの再起動を行うものである。そのとき、他の計算機上で該ソフトウェアを再起動すべきとき、ネットワークリンク内のいずれの計算機上で起動すべきかに関する情報をもつこと、また、該情報を生成する手段を有することを特徴とする。

【００７６】実施例8．この実施例は、障害発生時の主記憶上のデータのダンプの高速化を図る例である。主記憶を分割し、複数のネットワークリンクを通して同時に、複数の計算機の2次記憶上にダンプすることにより、処理の高速化を図る。これは、分割された主記憶領域に対して、それぞれ、ネットワークリンク、ダンプ先計算機、その計算機上のデータダンプ用の2次記憶領域を登録しておくことにより、達成される。

【００７７】本実施例は、図１５にあるような複数のネットワークで接続された複数計算機上で適用されるのであるが、高速主記憶データダンプは以下のように実現される。各計算機は図１６に示すような主記憶の管理表をそれぞれ持っている。図１６は、計算機0用の管理表を示している。０から１１までの主記憶領域は3つに分けて管理され、０から３の領域の主記憶データは計算機0のディスク装置0に（１４０１）、４から７の領域の主記憶データはネットワーク０を通して、計算機1のディスク装置0に（１４０２）、８から１１の主記憶領域のデータはネットワーク１を通して、計算機2のディスク装置0に（１４０３）対応づけられている。計算機に障害が発生したときは、リセット後の計算機立ち上げ時に、該管理表に従って主記憶データのダンプが行われる。また、システム機構は、図１５に示したが、図１７に示すシステムでもよい。

【００７８】このようにして、障害発生時になくてはならない、主記憶データの採取を高速に行うことができる。また、稼働率は、故障修理期間を短くすることによって向上できる。従って、システム再立ち上げを行う場合に、主記憶データのダンプ時間を短くすることは、稼働率向上に寄与する。

16

【００７９】以上のように、この実施例では、複数のネットワークリンクに複数台接続された計算機によって構成され、互助動作する計算機システムにおいて、該システム内のある計算機に障害が発生した時は、該計算機の主記憶内容を、あらかじめ情報設定手段によって設定されたコアダンプ情報によって分割し、定められたネットワークリンクを通して自身を含めた該ネットワークリンク内の計算機に送出することにより、退避することを特徴とする計算機システムについて説明した。

【００８０】実施例9．この実施例は、主記憶上のバッファにキャッシュされたディスクデータの一貫性を維持することを課題としており、これはシステム再立ち上げの場合、ディスク上に構築されたファイルシステムの一貫性回復のために要する時間を最小に押さえることを可能にする。すなわち、チェックサムによる主記憶上のディスクデータを検証し、自計算機上のディスク装置、および、複写されたデータを持つ他計算機のディスク装置に、ネットワークを通して書き込みを行うことにより実現される。

【００８１】この実施例は、図1に示すような計算機システムに適用され、１１５と１１６の関係にあるディスク装置が本実施例の対象である。すなわち、ディスク装置１１５のデータは、ディスク装置１１５に対して書き込みがあるたびにディスク装置１１６に複写される。計算機１０３上に読み出された、ディスク装置１１５のデータに対して、更新が加えられるとブロック毎にチェックサムデータがとられる。

【００８２】図１８は本実施例による、計算機上のディスクデータの管理情報を示したものである。ドライブーセクタとなっている項目は、正／副ディスク装置の計算機とドライブ番号を表しており、フラグは、各バッファのステートを示し、チェックサムデータには、各バッファデータ更新の度にチェックサムデータが格納される。また、図１９はステートの種類を示す図である。図１８では、１５０１において管理されるディスクデータは計算機Ａのディスク装置0で、データの複写先は、計算機Ｂのディスク装置0で、第0セクタのデータが格納されていることを意味する。このバッファのステートはＢＵＳＹでＣＰＵがデータ参照更新、もしくは、チェックサムデータ計算、書き込み中である。１５０２のバッファは未使用である。１５０３のバッファは計算機Ａのディスク装置0で、データの複写先は、計算機Ｂのディスク装置0で、第2セクタのデータが格納されている。このバッファのステートはＤＩＲＴＹで、ディスク装置に書き戻す必要のあるデータである。１５０４のバッファは未使用である。

17

【００８３】ひとたび、この状態で、計算機１０３に相当する計算機が障害を起こし使用不能になった時は、該計算機をリセット後、再立ち上げの際に、該管理表に基づき、ＢＵＳＹ、または、ＤＩＲＴＹであるバッファの内容に対しチェックサム計算を行い、バッファの内容から得られたチェックサムと管理表中のチェックサムデータと一致したならば、バッファ内容が正しいものとして正／副両方のディスクにバッファの内容の書き出しを行う。このようにして、障害発生時の主記憶上のバッファに読み出され変更されたデータもチェックサムにより、データが破壊されていないことがわかれば、ディスクに書き戻し、ディスクデータの一貫性を保つことができる。

【００８４】以上のように、この実施例は、１つ、あるいは、複数のネットワークリンクに複数台接続された計算機によって構成され、互助動作する計算機システムである。該計算機システムにおいて、ディスク記憶装置上のデータは異なる計算機上のディスク記憶装置に多重に格納されており、データとしてファイルシステム等が構築されており、該ディスクデータ利用時は該データは主記憶上のバッファに展開され、主記憶上バッファのデータは更新時にチェックサムを実行する。そのとき、該データを利用中の計算機が障害状態となり、該計算機の停止後、再起動時に主記憶上のバッファは展開されていたディスクデータで、更新済みで、かつ、チェックサムデータによりデータの正当性が認められ、さらに、ディスク装置への書き戻しが行われていないデータを、該計算機のディスク装置、および、データの多重化の行われている他計算機のディスク装置に書き戻すことを特徴とするものである。

【００８５】実施例１０．本実施例はディスク装置への書き込みの際にもしエラーが発生しても、書き込み前のデータ状態を保証するものである。エラーの要因としては、電源断、ディスクヘッドの損傷などが有り得る。本実施例の適用されたディスク装置は、データ書き込みを図２０のフローチャートに示す手順で行う。書き込み要求があれば、まず書き込み先のセクタのデータを退避する（１７０１）。退避先としてディスク領域を利用することも、他の不揮発性記憶を利用することも可能である。セクタデータは図２１に示すような形で退避される。すなわち、書き込み先のセクタ番号１６０１、書き込み操作が始まったことを示すＢＥＧＩＮマーク１６０２、そして、データ１６０３が退避される。このようにデータ退避が終了したならば、データの書き込みを始め、（１７０２）、データの更新に成功すると、１６０４のＣＯＭＭＩＴマークを書き入れる（１７０３）。仮に、データの退避中に障害がおきても、ディスクデータは保持される。また、データ更新中にエラーが発生した場合、退避データを調べればＣＯＭＭＩＴマークのないデータはデータ更新中であるとわかるので、この退避したデータにより元のデータの復旧が可能になる。また、退避データは不要になると開放する。

18

【００８６】以上で説明した機能は、データベースプログラムなどの、保証すべき信頼性が特に高いアプリケーションに利用されるべき機能で、全てのディスクに対する書き込みオペレーションに対し成功か不成功かの場合に、更新後、更新前の状態を必ず保証するディスク装置である。これにより、従来、アプリケーションプログラムの中で行っていたデータ一貫性保持操作が簡略化され、かつハードウェアレベルでデータ一貫性が保証されるため、システムの高速化も期待できる。

【００８７】以上のように、この実施例では、データベースプログラムなどの、厳密にデータの一貫性が要求されるプログラムに利用されるディスク記憶装置において、データ更新に当たって、２フェーズコミットメントを行うことにより、データ更新時に、該ディスク記憶装置に障害が発生しても、該更新操作前のデータを保証することを特徴とする例について述べた。

【００８８】

【発明の効果】第1の発明によれば、被検ソフトウェアおよび障害検出ソフトウェアは、それぞれシステムの負荷を検出し、その負荷に応じて検出信号の送受信間隔を調整するため、タイムアウトを使った障害発生の判断がより正確になる。また、双方のソフトウェアでそれぞれ負荷検出を行うことにより被検ソフトウェアで送信間隔を調整することに関する障害が発生した場合、障害検出ソフトウェア側で検出することができる効果がある。また、ハードウェアではなく、ソフトウェアで障害検知に関する機能を実現するため障害検知に要するコストを低減することができる。

【００８９】第2の発明によれば、被検ソフトウェアが検出信号の送出間隔を障害検出ソフトウェアに通知する。そのため、障害検出ソフトウェア側で送出間隔を調べるための処理を省くことができる。それにより、システムの負荷を減らすことができる。

【００９０】第3の発明によれば、被検ソフトウェア側に負荷検出部を持たせ、負荷により検出信号の送出間隔の調整を行う。また、その送出間隔の情報も検出信号中に含ませ障害検出ソフトウェア側に通知するので、障害検出ソフトウェア側では、負荷検出部を持つ必要がなく、また、その処理を行う必要がない。そのため、障害検出ソフトウェアのプログラムステップ数を減らすことができ、また、その処理を省くことができるため、システムの負荷を減らすことができる。

【００９１】第4の発明によれば、被検ソフトウェアがいくつかのプログラムのサービスを利用して成り立っている時、あるいは、相互にサービスを利用しあって成り立っている時、プログラム間の関係を示す管理情報を持つことによって、関連するすべてのプログラムを監視することができ、より正確な障害検出ができるという効果

がある。

【００９２】第5の発明によれば、障害検知機構を2重化することにより、障害検知機構自体の障害による、システム障害を回避することができる。

【００９３】第6の発明によれば、検出信号中に障害検出ソフトウェアで実行すべき動作手順を指示するメッセージを含ませ、障害検出ソフトウェア側でそのメッセージに対する手順情報を持つ。このため、障害検出ソフトウェア側で不必要な処理を省くことができ、的確な処理を行うことができる。

【００９４】第7の発明によれば、各計算機の状態を調査する調査プログラムを持つ。これにより、被検ソフトウェアに障害が検出され、かつ他の計算機への移行が必要となった場合、各計算機の負荷状況と資源状況を考慮して最適な移行先を選ぶことができる。

【００９５】第8の発明によれば、障害発生時に主記憶装置の内容を分割して、複数の他の計算機の2次記憶装置に退避することができる。そのため、主記憶装置のデータの採取を高速に行うことができ、主記憶装置のデータのダンプ時間を短くすることができる。このため、故障修理期間を短くすることができるので、稼働率向上に寄与する。

【００９６】第9の発明によれば、分割退避先を設定した管理表を備えているので、主記憶装置のデータの採取先をすみやかに決めることができ、主記憶装置のデータのダンプの高速化を図ることになり、稼働率向上に寄与するという効果がある。

【００９７】第１０の発明によれば、ネットワークを介して分割されたデータを退避する作業を他の計算機に分割することができるため、主記憶装置のデータのダンプの高速化を図ることができ、稼働率向上に寄与することができる。

【００９８】第１１の発明によれば、主記憶装置上に読み出され、変更されたデータも、障害発生後の回復時に、データの正当性を調べ2次記憶装置に書き戻す事ができる。そのため、ディスクデータの一貫性を保つことができ、システム再立ち上げの時、ディスク上に構築されたファイルシステムの一貫性回復のために要する時間を最小に押さえることができるので、稼働率向上に寄与する。

【００９９】第１２の発明によれば、多重系システムにおいても上記第１１の発明と同様の効果が得られる。

【０１００】第１３の発明において、ディスク装置への書き込みの際に、更新される前のデータを退避しておくためディスクへの書き込み命令が異常終了した場合にも、書き込み命令の発行以前のデータ状態を必ず保証することができる。このように、本発明は、データベースプログラムなどの、保証すべき信頼性が特に高いアプリケーションに利用されるべき機能で、全てのディスクに対する書き込みオペレーションに対し、成功か不成功か

の場合に、更新後、更新前の状態を必ず保証するディスク装置である。これにより、従来、アプリケーションプログラムの中で行っていた、データ一貫性保持操作が簡略化され、かつ、ハードウェアレベルでデータ一貫が保証されるため、システムの高速化も期待できる。

【図面の簡単な説明】

【図1】本発明を適用する計算機システムの例を示す図。

【図2】本発明を適用したソフトウェア構成例を示す図。

【図3】本発明における頻度表例を示す図。

【図4】本発明を適用したソフトウェア構成例を示す図。

【図5】本発明における頻度表例を示す図。

【図6】本発明を適用したソフトウェア構成例を示す図。

【図7】本発明における依存関係表例を示す図。

【図8】本発明を適用したソフトウェア構成例を示す図。

【図9】本発明における副障害検出プログラムの動作アルゴリズムを示す図。

【図１０】本発明を適用したソフトウェア構成例を示す図。

【図１１】本発明における障害に対する手順についての対応表の例を示す図。

【図１２】本発明におけるソフトウェア構成例を示す図。

【図１３】本発明における被検プログラムの実行条件データの例を表す図。

【図１４】本発明における計算機の負荷、資源状況の調査結果の例を示す図。

【図１５】本発明を適用する計算機システムの例を示す図。

【図１６】本発明における計算機0用の主記憶ダンプ先の管理表を示す図。

【図１７】本発明を適用する計算機システムの例を示す図。

【図１８】本発明における計算機上のディスクデータの管理情報を表す図。

【図１９】本発明におけるバッファのステートの種類を示す図。

【図２０】本発明におけるディスクデータ書き込みアルゴリズムを示す図。

【図２１】本発明における退避データ形態例を表す図。

【符号の説明】

１０３　計算機(主)
１０４　計算機(副)
１０５　スタンドバイサーバ
１０６　障害検出プログラム
１０７　障害検出プログラム(副)

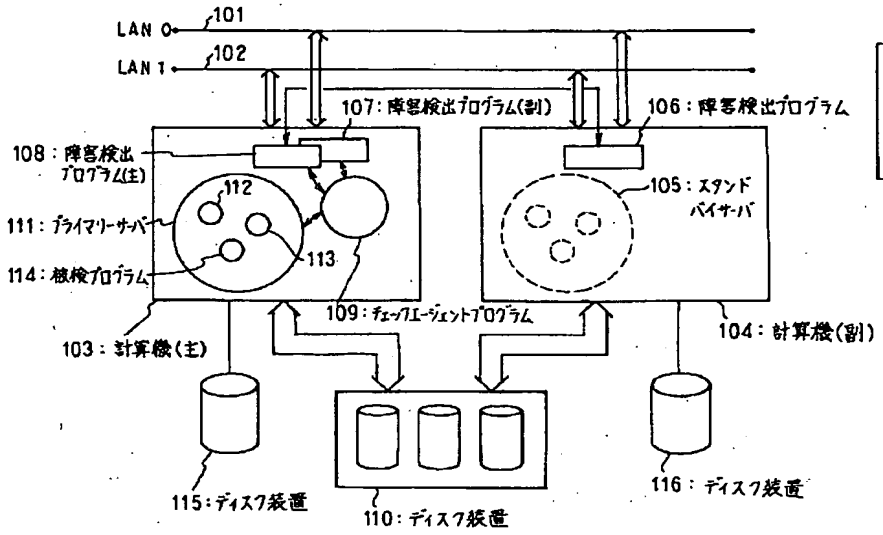
21

１０８　障害検出プログラム(主)
１０９　チェックエージェント プログラム
１１０，１１５，１１６　ディスク装置
１１１　プライマリ ーサーバ
１１２，１１３，１１４　被検プログラム

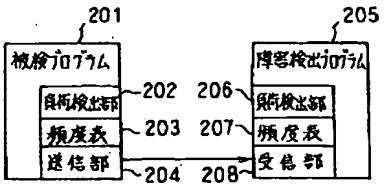
22

２０２，２０６　負荷検出部
２０３，２０７　頻度表
２０４　送信部
２０８　受信部

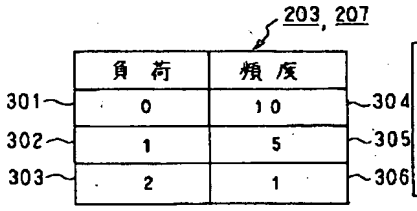
【 図1 】



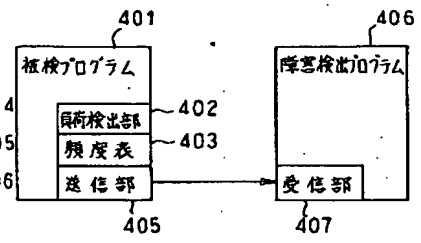
【 図2 】



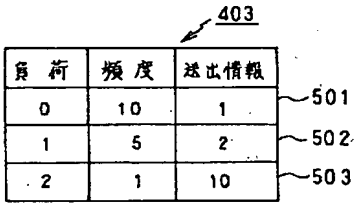
【 図3 】



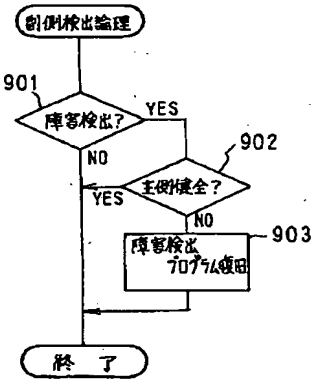
【 図4 】



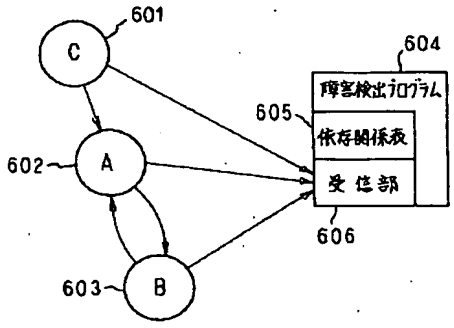
【 図5 】



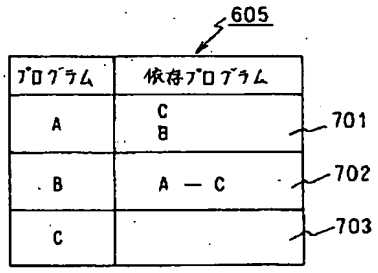
【 図9 】



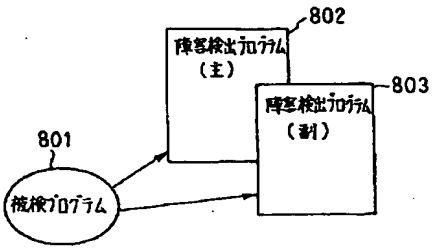
【 図6 】



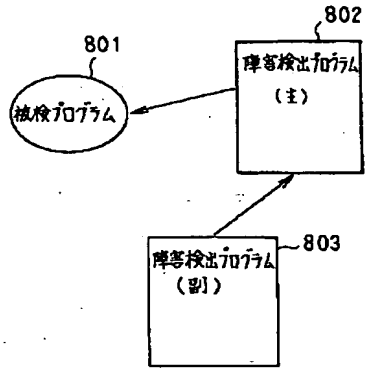
【 図7 】



【 図8 】



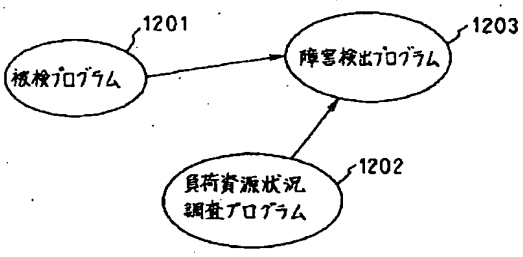
【 図１０ 】



【 図１１ 】

メッセージ	手　順	
正　常	何もしない	1001
停　止	タイムアウト延期	1002
開　始	監視開始	1003
終　了	終了処理	1004
障害１	３回リトライ	1005
障害２	ディスクデータを修復	1006
障害３	他計算機で再実行	1007

【 図１２ 】



【 図１３ 】

プログラム	負　荷	I/O頻度	メモリ	
A	1	100	2	1101
B	4	1000	0.1	1102

【 図１４ 】

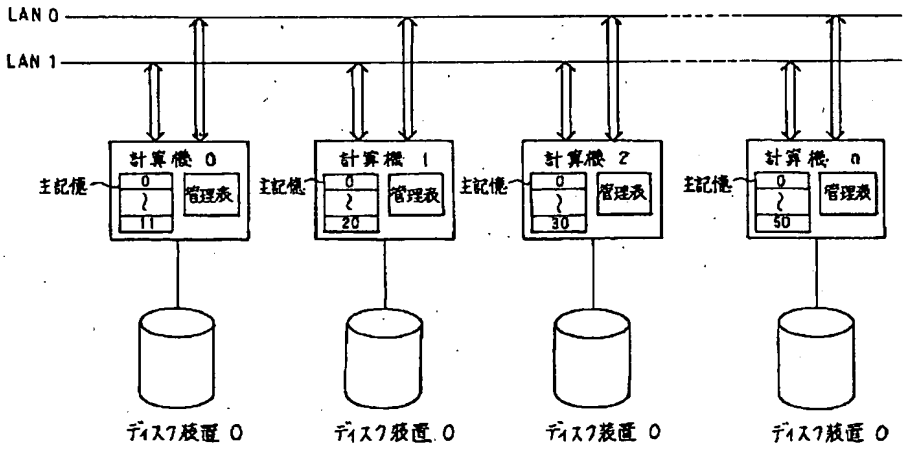
計算機名	負荷	I/O頻度	メモリ	
A	0.1	10	100	1301
B	2	50	10	1302
C	1	1000	50	1303

【 図１６ 】

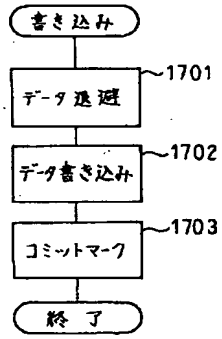
計算機 0 用の主記憶ダンプ先の管理表

主記憶領域	ネットワーク	計算機	ディスク	
0 ～ 3	—	0	disk 0	1401
4 ～ 7	LAN 0	1	disk 0	1402
8 ～11	LAN 1	2	disk 0	1403

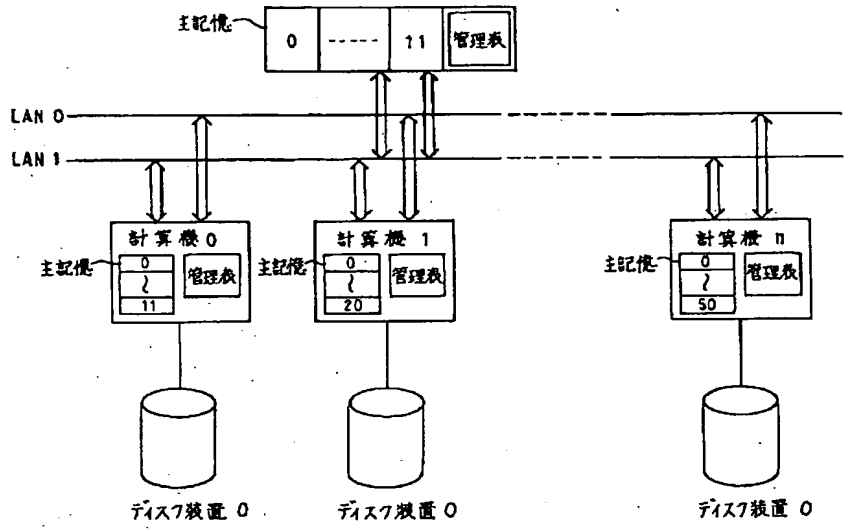
【 図１５ 】



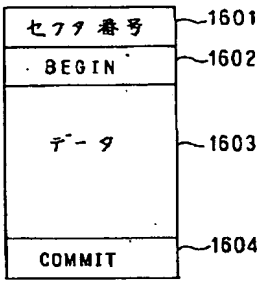
【 図２０ 】



【 図１７ 】



【 図２１ 】



【 図１８ 】

ドライブ-セクタ	フラグ	チェックサムデータ	
A0, B0, 0	BUSY	×××××	1501
A0, B0, 1	FREE	×××××	1502
A0, B0, 2	DIRTY	×××××	1503
A0, B0, 3	FREE	×××××	1504

【 図１９ 】